

ATLAS
MANUAL DE USUARIO
COMPONENTES DE DOCUMENTUM
IMPORTAR

Versión 1.1

Área de Aplicaciones Especiales y Arquitectura de Software



Agencia de Informática y Comunicaciones de la Comunidad de Madrid



Hoja de Control

Título	Manual de usuario Componentes de Documentum. Importar.		
Documento de Referencia	NORMATIVA ATLAS		
Responsable	Área de Aplicaciones Especiales y Arquitectura de Software		
Versión	1.1	**Fecha Versión**	26/10/2012

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.1	Versión inicial del documento	Área de Aplicaciones Especiales y Arquitectura de Software	21/11/2011
1.2	Correcciones ortográficas	Área de Aplicaciones Especiales y Arquitectura de Software	26/10/2011

Índice

1. INTRODUCCIÓN

Este documento contiene el manual de uso del componente visual ***Importar*** del Framework Atlas. En él se incluye información sobre cómo utilizar dicho componente en una aplicación Web, así como información acerca de la configuración de los parámetros fundamentales del componente.

1.1. AUDIENCIA OBJETIVO

Este documento está orientado a toda aquella persona que esté desarrollando una aplicación Web basada en el Framework Atlas y necesite utilizar componentes de presentación en su aplicación Web.

1.2. CONOCIMIENTOS PREVIOS

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

- Java Server Faces (JSF)
- Facelets
- Spring Framework
- Hibernate
- Gestor documental Documentum

Para saber más sobre dichas tecnologías, consultar el apartado de este documento, *Enlaces Relacionados*.

2. DESCRIPCIÓN

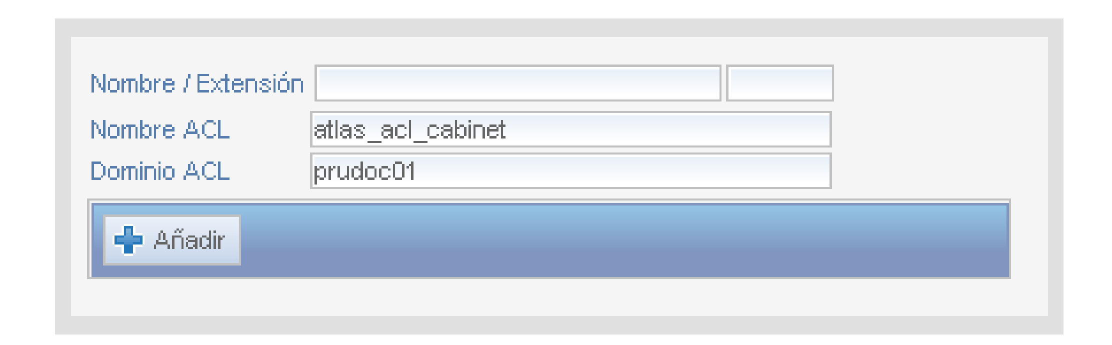
Este componente permite introducir y modificar documentos en el sistema de gestión documental.

Las características principales de este componente son:

- Soporte para texto localizado, pudiendo forzar, en caso de ser necesario, una localización diferente a la establecida por el navegador.
- Posibilidad de mostrar mensajes de información y error en cada ejecución del componente.
- Posibilidad de personalizar los estilos del componente.
- A través de un único componente de importación tenemos la capacidad de importar cuantos documentos consideremos oportunos.
- A través de un único componente de modificación podemos modificar un mismo documento las veces que hagan falta.

Este componente puede ser extendido por el desarrollador para recoger y guardar los metadatos del documento

que considere oportunos. Aparte de los campos de texto asociados, el componente mostrará campos de texto para el nombre del fichero, la extensión del mismo y el nombre y dominio ACL y un control para subida del documento a importar o modificar. El aspecto del componente está sujeto a la implementación particular de cada programador.



3. INSTALACIÓN Y CONFIGURACIÓN

En este apartado se incluye información sobre la instalación y la configuración del componente *Importar.*

3.1. INSTALACIÓN

El componente *Importar* ya viene instalado en el arquetipo web documentum, incluido con el módulo de componentes visuales. Por este motivo no es necesaria una instalación adicional si se parte del arquetipo.

3.2. CONFIGURACIÓN

No es necesaria ninguna configuración adicional si se ha partido del arquetipo web documentum.

4. USO

Una vez instalado el módulo de componentes puede procederse a su utilización. Para ello deben realizarse los pasos indicados en los siguientes apartados:

4.1. Paso 1: Definición del espacio de nombres de etiquetas de Documentum

Es necesario crear un fichero *xhtml* y establecer la definición del espacio de nombres para las etiquetas de componentes de Documentum. Un ejemplo de cabecera de fichero *xhtml* es la siguiente:

Cabecera de fichero xhtml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:gdoc="http://atlas.core.componentes/jsf/gdoc"
      xmlns:a4j="http://richfaces.org/a4j">
```

4.2. Paso 2: Inserción en la página de la etiqueta de Atlas

Hay que distinguir dos opciones de inclusión del componente: importación y modificación, estaremos en uno u otro caso si el atributo idDocumento es nulo (importación) o no (modificación).

Se incluirá una etiqueta gdoc:importar que inserta el componente en la página. A continuación se muestra un ejemplo:

Ejemplo.xhtml

```
<gdoc:importar backingBean="impModDocBacking"
    documentType="dm_document"
    idDocumento="#{documentosBean.documento.idDoc}"
    listaAtributos="r_object_id"
    rutaRepositorio="/prudoc01" />
```

Los atributos del componente son los siguientes:

Nombre atributo	Obligatorio	Descripción
backingBean	SI	Id del backing bean asociado, debe corresponder a una clase de tipo *atlas.gdoc.jsf.JsfImpModDocBean*. Si no se especifica se utiliza el definido por defecto en el arquetipo.
documentType	SI	Tipo de documento a modificar o importar.
listaAtributos	SI	Lista de atributos del documento a modificar o insertar. Debe haber al menos un atributo válido.
idDocumento	SI	Id del documento. Se debe especificar en la modificación.
rutaRepositorio	NO	En importación es obligatoria e indica la ruta donde se va a almacenar el documento. En modificación es opcional.
nombre	NO	Nombre para el fichero, si se especifica no será posible modificarlo.
extensión	NO	Extensión para el fichero en caso de utilizar un nombre fijo.
actualizarNombre	NO	Variable de tipo boolean. Si es true se utilizar para crear o modificar el fichero el nombre y extensión del fichero que se suba, si es false se utilizarán el nombre y extensión proporcionados por el usuario. Su valor por defecto es false.
mostrarACL	NO	Propiedad de tipo boolean. Si es true se muestran los campos nombre ACL y dominio ACL. Su valor por defecto es true.
render	NO	Lista de ids de componentes que se deben renderizar al importar o modificar un componente.
title	NO	Título para el componente.
locale	NO	El Locale asociado al componente en caso de que no nos interese el locale resuelto a partir de la configuración del navegador.

class	NO	Atributo class de la capa div que contiene al componente.
style	NO	Atributo style de la capa div que contiene al componente.
rendered	NO	Atributo booleano que permite mostrar u ocultar el componente.

El componente no se mostrará y volcará un mensaje de error en el FacesContext en los siguientes casos:

- No se proporciona alguno de los siguientes atributos: tipoDocumental, backingBean y atributosDetalle.
- La lista de atributos está vacía.
- El valor del atributo backingBean, no está asociado a ningún backingBean.
- En el caso de importación, si el atributo rutaRepositorio no ha sido proporcionado.

Cuando el componente interpreta la función de importación, tras importar un documento satisfactoriamente, el componente se recarga con todos los controles para poder volver a importar otro documento y así sucesivamente.

Cuando el componente interpreta la función de modificación, tras modificar un documento satisfactoriamente, el componente se recarga con los valores actualizados de los atributos y se encuentra disponible para otra modificación.

4.3. RECOMENDACIONES Y BUENAS PRÁCTICAS

El desarrollador debe crear una implementación de la interfaz *importarModificarService* en el paquete *atlas.gdoc.jsf.facelets.service.* Deberá implementar dos métodos:

1. *createDynamicFields*: creará los controles necesarios del formulario para recoger los metadatos del documento. Este método recibe dos parámetros fundamentales: un mapa denominado *atributosMap* cuyas keys corresponden a los atributos indicados en el componente y el nombre del backing bean asociado al componentes denominado *backingName* para poder definir los bindings de los componentes JSF con las propiedades del backing bean, que en este caso son *atributosMap* y *atributosList*. Cada entrada del *atributosMap* contendrá el nombre (key) y valor (value) de los atributos. La variable *atributosList* va a contener objetos Atributo que estarán asociados normalmente a campos de texto a través de bindings de JSF.
2. *loadDataToFacesContext*: Tiene dos tareas fundamentales:
    a. Importar o modificar un documento en caso de ser necesario, es decir, cuando detectemos que el usuario ha pulsado un botón.
    b. Cargar los datos asociados al documento como los atributos o los datos asociados del ACL, cuando se trate de una operación de modificación.

En caso de necesitar mostrar varios componentes, es necesario declarar en la configuración de JSF un backing bean por cada componente y asociarlo a cada componente mediante la propiedad *backingBean*.

4.4. EJEMPLO DE USO

Se puede ver un ejemplo del componente en la aplicación de integración de Atlas, en la sección de Documentum.

5. PREGUNTAS MÁS FRECUENTES

En este apartado se incluyen una lista de preguntas más frecuentes sobre el componente.

Pregunta: ¿Dónde puedo encontrar información general sobre los componentes?
Respuesta: En la aplicación de ejemplos de integración del Framework Atlas. En la sección de Documentum.

Pregunta: ¿Cómo se ha implementado el componente en su capa de presentación?
Respuesta: Mediante componentes de Facelets.

Pregunta: ¿Cómo puedo modificar los estilos del componente?
Respuesta: Mediante los atributos de estilo del componente. Para más información consultar la tabla de atributos en la sección de *Uso* del componente.

6. ENLACES RELACIONADOS

Producto	URL
Ajax4JSF	http://www.jboss.org/jbossrichfaces/
Barbecue	http://barbecue.sourceforge.net/
Commons BeanUtils	commons.apache.org/beanutils/
Commons Configurations	http://commons.apache.org/configuration/
Facelets	https://facelets.dev.java.net/
Hibernate	http://www.hibernate.org/
Hibernate Annotations	http://www.hibernate.org/hib_docs/annotations/reference/en/html_single/
JAXB	http://java.sun.com/webservices/jaxb/
Jcaptcha	jcaptcha.sourceforge.net/
JPA	http://java.sun.com/developer/technicalArticles/J2EE/jpa/
JSF	http://java.sun.com/javaee/javaserverfaces/
JSFUnit	http://www.jboss.org/jsfunit/
Log4J	http://logging.apache.org/log4j/
MyFaces Core	http://myfaces.apache.org/
RichFaces	http://www.jboss.org/jbossrichfaces/
Spring	http://www.springframework.org/
Spring Security	http://www.springframework.org/
Tomahawk	http://myfaces.apache.org/tomahawk/
Velocity	http://velocity.apache.org/